

**In the Claims:**

Please amend claims 1, 14, 15, and 26-36 as indicated below.

1. (Currently amended) A system, comprising:

one or more processors;

memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement a transaction manager, wherein the transaction manager is configured to:

manage a plurality of transactions initiated by one or more applications, wherein each transaction comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction;

pause currently executing ones of the plurality of transactions managed by the transaction manager in response to a pause request to pause the transaction manager, wherein while paused, the transaction manager does not allow any of the plurality of transactions managed by the transaction manager to complete; and

resume execution of the paused ones of the plurality of transactions managed by the transaction manager in response to a resume request.

2. (Previously presented) The system as recited in claim 1, wherein the transaction manager is configured to change the state of each of the plurality of transaction managed by the transaction manager.

3. (Previously presented) The system as recited in claim 2, wherein the transaction manager is configured to request permission to change the state of each of the plurality of transactions managed by the transaction manager prior to changing the state of each respective transaction.

4. (Previously presented) The system as recited in claim 3, wherein the transaction manager is configured to wait for permission to change the state of each of the plurality of transactions managed by the transaction manager.

5. (Previously presented) The system as recited in claim 1, wherein the transaction manager is configured to prohibit a change of state of each of the plurality of transactions managed by the transaction manager while the transaction manager is paused.

6. (Previously presented) The system as recited in claim 1, wherein the transaction manager is configured to support the execution of each of the plurality of transactions managed by the transaction manager within a respective current state while the transaction manager is paused.

7. (Previously presented) The system as recited in claim 1, wherein the system is configured to continue the execution of the one or more applications while the transaction manager is paused, except for changing of transaction states.

8. (Original) The system as recited in claim 1, wherein while the transaction manager is paused, the system is configured to perform operations on one or more individual transactions.

9. (Original) The system as recited in claim 8, wherein the operations comprise one or more from the following: rollback, abort, partial rollback, add/remove participant, and commit.

10. (Original) The system as recited in claim 1, wherein one or more of the transactions are local transactions.

11. (Original) The system as recited in claim 10, wherein while paused, the transaction manager is not allowed to change the state of the one or more transactions to the committing state.

12. (Original) The system as recited in claim 1, wherein one or more of the transactions are global transactions.

13. (Original) The system as recited in claim 12, wherein while paused, the transaction manager is not allowed to change the state of the one or more transactions to the preparing state nor to the committing state.

14. (Currently amended) A system, comprising:

one or more processors; and

memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement one or more application servers, wherein each one or more application servers is configured to:

run one or more applications each configured to initiate one or more transactions, wherein each of the one or more transactions comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for the transaction; and

provide one or more transaction managers configured to manage the one or more transactions initiated by the one or more applications,

wherein one of the transaction managers is configured to pause a corresponding one or more of the currently executing transactions in response to a pause request and to resume the corresponding one or more of the paused transactions in response to a resume request, wherein while paused, the transaction manager does not allow the corresponding one or more transactions to complete.

15. (Currently amended) A computer-implemented method, comprising:

using one or more computers to perform:

generating a request to pause a transaction manager;

pausing the transaction manager in response to said request, wherein while the transaction manager is paused, transactions currently executing and managed by the transaction manager are prohibited from completing;

generating a request to resume the transaction manager; and

resuming the transaction manager in response to said request, wherein when the transaction manager is resumed, paused transactions managed by the transaction manager are allowed to complete, wherein each transaction managed by the transaction manager comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction.

16. (Previously presented) The method as recited in claim 15, wherein said pausing comprises prohibiting the transaction manager from changing the state of the

transactions, wherein the transaction manager attempts to perform a state change on a transaction in response to input to the transaction manager.

17. (Original) The method as recited in claim 16, wherein the input comprises notification that an application has initiated a transaction.

18. (Original) The method as recited in claim 17, wherein said state change comprises a change from a nonexistent state to an active state.

19. (Original) The method as recited in claim 16, wherein the input comprises notification that an application has executed a commit transaction command.

20. (Original) The method as recited in claim 19, wherein said state change comprises a change from an active state to a preparing state.

21. (Original) The method as recited in claim 16, wherein the input comprises notification that all participants are prepared commit the transaction.

22. (Original) The method as recited in claim 21, wherein said state change comprises a change from a preparing state to a committing state.

23. (Original) The method as recited in claim 16, wherein the input comprises notification that all participants have committed the transaction.

24. (Original) The method as recited in claim 23, wherein said state change comprises a change from a committing state to a nonexistent state.

25. (Original) The method as recited in claim 16, further comprising, while the transaction manager is paused, continuing to support the execution of the one or more applications, except for the changing of transaction states.

26. (Currently amended) A computer-readable ~~accessible~~ storage medium[[,]] ~~comprising~~ storing program instructions, wherein the program instructions are computer-executable to:

generate a request to pause a transaction manager;

pause a plurality of transactions managed by the transaction manager in response to said request to pause the transaction manager, wherein while the transaction manager is paused, the plurality of currently executing transactions managed by the transaction manager are prohibited from completing, wherein each transaction comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction;

generate a request to resume the transaction manager; and

resume the plurality of transactions managed by the transaction manager in response to said resume request, wherein when the transaction manager is resumed, the plurality of paused transactions managed by the transaction manager are allowed to complete.

27. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 26, wherein the program instructions are computer-executable to pause the transaction manager by prohibiting the transaction manager from changing the state of any of the plurality of transactions managed by the transaction manager, wherein the transaction manager attempts to perform a state change on a transaction in response to input to the transaction manager.

28. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 27, wherein the input comprises notification that an application has initiated a transaction.

29. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 28, wherein said state change comprises a change from a nonexistent state to an active state.

30. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 27, wherein the input comprises notification that an application has executed a commit transaction command.

31. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 30, wherein said state change comprises a change from an active state to a preparing state.

32. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 27, wherein the input comprises notification that all participants are prepared commit the transaction.

33. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 32, wherein said state change comprises a change from a preparing state to a committing state.

34. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 27, wherein the input comprises notification that all participants have committed the transaction.

35. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 34, wherein said state change comprises a change from a committing state to a nonexistent state.

36. (Previously presented) The computer-readable ~~accessible~~ storage medium as recited in claim 27, wherein the program instructions are computer-executable to,

while the plurality of transactions managed by the transaction manager ~~is~~ are paused, continue to support the execution of the one or more applications, except for the changing of transaction states.